Amendments to the Specification:

Please replace the paragraph beginning at page 10, line 1 with the following amended paragraph:

An individual instruction may be initially split among the plurality of 16-bit buffers 520-555. For example, a 64-bit instruction may begin in the third buffer 530 and end in the sixth buffer 545. The alignment mux of Figure 5 ensures that this instruction is aligned prior to processing in the pipeline by transferring the data to the plurality of multiplexers 560-575 and then selecting the proper data to send to the flops 585-592. To reduce the risk that bubbles are [[not]] inserted in the pipeline, the alignment mux reloads buffers 510, 515 in the memory 505 once all the instruction data in the buffers 510, 515 is dispatched to the pipeline. In one embodiment, the memory 505 may be a cache memory.

Please replace the paragraph beginning at page 12, line 7 with the following amended paragraph:

A multiplexer 628 receives a branch target address and the current state (cstate) of the instruction position in the buffers 510, 515. The multiplexer 628 selects either the branch target address or the current state to load the flop 630 with either the current state ~~of~~ or the branch target address, which may become the current state. The current state is then combined with the width of the current instruction in block 625. By combining the current state with the current width, the position of the beginning of the next instruction (nstate) may be determined. This position information is then supplied to a second alignment multiplexer 635 to be used in aligning the next instruction. The next instruction position information may also feed back to the flop 630 at the next clock cycle. At the next clock cycle, the next instruction position becomes the current instruction position, and this information is updated in the flop 630. The next instruction position is also supplied to the transition block 645 to aid in determining is a transition from the first buffer 510 to the second buffer 515 has occurred.

Please replace the paragraph beginning at page 13, line 15 with the following amended paragraph:

The transition logic determines whether either of the buffers 510 and 515 will be emptied after processing of the next instruction. The transition block 645 includes the next state position and the next instruction width as inputs. The transition block 645 then determines based on the next state position and instruction width whether either of the buffers 510 and 515 will be exhausted after the next instruction. For example, if the next state position indicated is the beginning of the 16-bit buffer 530 and the next instruction width is 64-bits, the transition block 645 determines that the instruction will be taken from the 16-bit buffers 530, 535, 540, and 545, thus completely emptying the first buffer 510. The transition block 645 may then send a signal to the flop 650 indicating that the first buffer 510 is available to be reloaded, which may generate a ~~requires~~ request to the memory to fill the empty buffer 510.